

НУБІП України

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

15.03 – КМР.1862 – “С” 2021.11.01.05 ПЗ
НУБІП України

ЗАХАРЧЕНКА МИХАЙЛА ВОЛОДМИРОВИЧА

2022 р.
НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

НУБІП України

Факультет інформаційних технологій

НУБІП України

УДК
«ПОГОДЖЕНО»
Декан факультету

«ДОНУСКАЄТЬСЯ ДО ЗАХИСТУ»
Завідувач кафедри комп'ютерних наук

інформаційних технологій

НУБІП України

Глазунова О.Г., д.п.н., професор

Голуб Б.Л., к.т.н., доцент

202_р.

202_р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

НУБІП України

на тему Дорадча WEB-система діагностики захворювання
свійських тварин

Спеціальність 122 Комп'ютерні науки

(код і назва)

Освітня програма Комп'ютерно еколого-економічний моніторинг

(назва)

Орієнтація освітньої програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми

НУБІП України

(науковий ступінь та вчене звання)

(підпис)

(ПІБ)

Керівник магістерської кваліфікаційної роботи

К. Т. Н., доцент

(науковий ступінь та вчене звання)

Бородкін Георгій Олексійович

(підпис)

(ПІБ)

К. Т. Н., доцент

(науковий ступінь та вчене звання)

Бородкіна Ірина Лаврентіївна

(підпис)

(ПІБ)

Виконав

(підпис)

Захарченко Михайло Володимирович

(ПІБ студента)

КИЇВ-2022

НУБІП України

ВСТУП

1. СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	6
--	---

НУБІП України

1.1 Постановка завдання

1.1 Постановка завдання	6
-------------------------------	---

1.2 Дослідження предметної області	7
--	---

1.3 Аналіз наявних рішень	8
---------------------------------	---

1.3.1 Основи клінічної діагностики	8
--	---

НУБІП України

1.3.2 Класифікація хвороб

1.3.2 Класифікація хвороб	10
---------------------------------	----

1.3.3 Симптоми хвороби: їхня класифікація, розпізнавання та оцінка діагностичної значущості	10
---	----

1.3.4 Діагноз хвороби та його класифікація	12
--	----

НУБІП України

2. МОДЕЛЮВАННЯ СИСТЕМИ

2. МОДЕЛЮВАННЯ СИСТЕМИ	14
------------------------------	----

2.1 Діаграма прецедентів	14
--------------------------------	----

2.2 SADT діаграма	16
-------------------------	----

2.2.1 Діаграма послідовності	18
------------------------------------	----

НУБІП України

2.3 Діаграма активності

2.3 Діаграма активності	20
-------------------------------	----

2.4 Діаграма кооперації	21
-------------------------------	----

2.5 Діаграма діяльності	22
-------------------------------	----

НУБІП України

3. РОЗРОБКА СИСТЕМИ

3. РОЗРОБКА СИСТЕМИ	24
---------------------------	----

3.1 Архітектура системи	24
-------------------------------	----

3.2 Інформаційне забезпечення	26
-------------------------------------	----

3.3 Засоби розробки Web-сайту	31
-------------------------------------	----

НУБІП України

3.3.1 Мова розмітки гіпертекстових сторінок HTML

3.3.1 Мова розмітки гіпертекстових сторінок HTML	31
--	----

3.3.2 Каскадна таблиця стилів CSS	32
---	----

3.3.3	Препроцесор гіпертексту PHP	34
3.3.4	SQL	35
3.4	Тестування системи	37

	ВИСНОВКИ	43
--	----------	----

	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	44
	ДОДАТОК А	46
	ДОДАТОК Б	49

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

ВСТУП

Актуальність теми. Зараз розвиток тваринницької галузі є одним із головних завдань сільського господарства. Держава має високий потенціал у сільськогосподарській сфері, оскільки вона має відповідні природно-кліматичні умови і пасовищні угіддя, які сприяють успішному розвитку тваринництва.

Величезну роль у продуктивності свійських тварин має вплив різних захворювань. Ці фактори є серйозною перешкодою для здорового та сталого тваринництва. У свійських тварин, зокрема корів, є низка певних особливостей у діагностуванні захворювань і постановці діагнозу. З огляду на світовий досвід з розробки інтелектуальних систем зроблено висновки щодо їх створення.

Унікальність цієї системи визначається використовуваними методами розробки та базою знань. Мета дослідження полягає в тому, щоб розробити автоматизований інструмент для діагностики захворювань свійських тварин і вивчити його вплив на діагностику хворіб худоби.

Мета роботи. Полягала у вивченні та розробці експертної системи для ефективного використання в діагностиці захворювань сільськогосподарських тварин. Було розроблено платформу для надання допомоги в діагностиці захворювань. Клінічне застосування цієї системи допоміжної діагностики підвищує точність розпізнавання захворювань. Дана програма з діагностики захворювань використовує внесену інформацію, далі її систематизують, узагальнюючи знання про захворювання у фахових підручниках, медичних звітах і дослідницьких документах, а також інформацію про лікування. Під час проведеного дослідження було успішно реалізовано допоміжну діагностичну експертну систему при захворюваннях свійських тварин. Використання цієї системи дасть змогу своєчасно діагностувати захворювання і призначити оптимальне лікування, тим самим знизити економічні втрати у фермерських господарствах.

1. СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Нюстановка завдання

Метою розробки є полегшення передання знань провідних спеціалістів з лікування тварин та центральних ветеринарних організацій ветлікарям-практикам, концентрації та постійного оновлення довідкових відомостей, необхідних для їхньої ефективної роботи; надання можливості діагностування хвороб і отримання рекомендацій щодо лікування тварин будь-яким користувачем.

Система призначена для автоматизованої діагностики хвороб тварин, видачі рекомендацій щодо профілактики хвороб, лікування тварин та оздоровлення, формування інформаційних довідок щодо хвороб, ознак хвороб, збудників, широти розповсюдження хвороб, вражаючих систем, довідкової літератури.

Експертна система інформаційно-методичного центру слугує досягненню таких цілей:

- Актуалізація науково-методичних знань щодо хвороб тварин і заходів боротьби з хворобами
- Актуалізація даних про засоби боротьби з хворобами
- Передача знань і даних в експертні системи підприємств
- Консультаційне обслуговування користувачів системи, фермерів, власників тварин.

На рівні підприємства адміністратор зі знань і даних обирає ті, що є актуальними для підприємства, яке він обслуговує; доповнює та коригує їх з урахуванням місцевих умов і свого досвіду.

Використання експертних систем ветеринарним персоналом середньої ланки значно розширює їхні можливості як спеціалістів.

Для роботи з описуваною програмою достатньо мати найзагальніші знання комп'ютера, оскільки в програмах використовуються прості діалогові засоби.

Під час проектування і розроблення ЕС необхідно відповісти на низку таких запитань:

- Які способи подання знань найкраще використовувати?
- Який механізм логічного виведення краще підходить для отримання найбільш правильного рішення?

- Як користувачі мають взаємодіяти з ЕС?
- Які сервісні функції мають бути реалізовані в ЕС?
- Які інструментальні засоби найкраще використовувати під час реалізації ЕС? тощо.

У системах, заснованих на знаннях, правила (або евристичні), за якими розв'язують проблеми в конкретній предметній області, зберігаються в базі знань. Проблеми ставляться перед системою у вигляді сукупності фактів, що описують деяку ситуацію, і система за допомогою бази знань виводить висновок із цих фактів.

1.2 Дослідження предметної області

Під час дослідження предметної області потрібно було вивчити й систематизувати масу інформації про захворювання свійських тварин, причини цих захворювань, симптоми, методи їхньої профілактики та лікування. Під час роботи з таким великим набором понять, необхідно дати їм чіткі визначення, щоб не виникло невизначеностей.

Симптом - у медичному розумінні це зовнішня ознака, за якою можна ідентифікувати захворювання.

Діагноз - встановлення хвороби на підставі дослідження хворого (у разі ЕС - на підставі отриманих симптомів).

Профілактика - сукупність заходів, що запобігають захворюванням.

Лікування - процес, бажаною (але не завжди досяжною) метою якого є поліпшення, зняття або усунення причин чи симптомів і проявів того чи іншого захворювання, патологічного стану або іншого порушення життєдіяльності, нормалізація порушених процесів життєдіяльності та одужання, відновлення

здоров'я.

Після визначення основних понять, було проведено спеціальну літератури на предмет виявлення симптомів, які однозначно визначають той чи інший діагноз. Під час проведення дослідження предметної області

застосовували об'єктно-орієнтовану декомпозицію, яка багато в чому визначила

й інструментальний засіб, і структуру бази знань. Під час використання

об'єктно-орієнтованого підходу вивчаються складові предметної області на предмет відносин між ними, наприклад на рисунку 1 зображено роботу

алгоритму.



Рис. 1 Робота алгоритму

1.3 Аналіз наявних рішень

1.3.1 Основи клінічної діагностики

Клінічна діагностика (грец. klinicalis - мистецтво лікування і diag- nosticon - здатний розпізнавати) - наука про розпізнавання хвороб тварин.

Іноді клінічну діагностику називають пропедевтикою (грец. propaideuo - попередньо навчак), оскільки вона дає основу для подальшого вивчення внутрішніх незаразних хвороб, епізоотології, паразитології, акушерства, хірургії

тощо дисциплін. Природно, освоєння клінічної діагностики неможливе без знання анатомії, фізіології, біохімії, мікробіології, патфізіології та ін. предметів. Клінічна діагностика складається з трьох основних розділів, взаємопов'язаних між собою:

1. Методи спостереження і дослідження тварини, тобто цей розділ охоплює техніку і порядок використання методів під час дослідження окремих систем і органів, корістко його можна назвати "лікарська техніка".

2. Вивчення виявлених ознак, їх відхилення від показників, які мають бути у здорових тварин. Тобто цей розділ вивчає симптоми і називається семіологією (семіотикою), від грец. semeion - ознака.

3. Особливості мислення лікаря під час розпізнавання захворювання - методика діагнозу.

Діагностичний процес починається з оволодіння методами спостереження та дослідження хворої тварини. Наскільки фахівець успішно оволодів цими діагностичними прийомами, настільки він буде успішно виявляти ознаки хвороби - симптоми. На підставі багатоговікового досвіду створено схему обстеження хворої тварини, що дає змогу отримати найповнішу інформацію про стан організму та про ознаки хвороби. З цією метою використовуються численні й різноманітні прийоми та методи, які постійно збагачуються й удосконалюються.

Необхідну для діагностики інформацію лікар отримує з розповіді власника тварини або від осіб, які доглядають за нею, під час об'єктивного дослідження пацієнта за допомогою різних методів (загальних, інструментальних, лабораторних тощо). Отримані при цьому відомості фіксують у спеціальних документах: журналах обліку амбулаторно або стаціонарно хворих тварин, в історії хвороби або диспансерній карті. При цьому дедалі ширшого застосування для запису інформації набувають ПЕОМ або комп'ютери. Їх використання допомагає не тільки зберігати інформацію про хворих тварин, а й застосовувати математичні методи діагностики патологічних процесів.

1.3.2 Класифікація хвороб

Нозологія (грец. nosos - хвороба і logos - вчення) - вчення про хворобу.

Поділяється нозологія на загальну та приватну. Загальна є теоретичною основою ветеринарної медицини і містить у собі: вчення про сутність хвороби; питання етіології, патогенезу, діагнозу, терапії та профілактики хвороб; питання їхньої класифікації та номенклатури. Приватна нозологія розглядає захворювання окремих систем і органів.

Класифікація хвороб - це їх групування за певним принципом. На основі цієї чи іншої класифікації складається перелік найменувань окремих хвороб або номенклатура хвороб. Сучасна класифікація та номенклатура хвороб тварин є результатом їх тривалого вивчення. Може бути видовою, віковою, етіологічною, системно-органною тощо.

1.3.3 Симптоми хвороби: їхня класифікація, розпізнавання та оцінка діагностичної значущості

Як уже згадували, симптоми хвороб вивчає семіологія (семіотика) - наука про симптоми та лабораторні показники хвороби. Поряд із наведеними вище визначеннями симптому, для ветеринарії загальноприйнятим є таке: "Симптом - усі прояви хвороби, в основі яких лежать функціональні та анатомічні зміни в органах і системах, що встановлюються клінічним дослідженням і відрізняють хвору тварину від здорової".

За клінічним проявом симптоми поділяють на:

- патогномонічні (специфічні) і випадкові. Патогномонічні (грец. pathos - страждання і gnosis - вказувати) - такі, що безумовно вказують на певну хворобу (наявність тілець Бабеші-Негрі - сказ; виявлення Babesia в еритроцитах - бабезіоз). Випадкові симптоми не властиві даній хворобі (жовтяниця при гастриті).

- типові (характерні) і нетипові (нехарактерні). Типові практично завжди зустрічаються при цьому захворюванні (тупий звук при крупозній пневмонії),

але можуть виявлятися і при інших патологічних станах (ексудативний плеврит).

- важливі та маловажні. Важливі - симптоми, на підставі яких робиться висновок, тобто важливі під час постановки діагнозу (гіпотермія, холодний липкий піт, частий ниткоподібний пульс - за умови розриву шлунку або

кишківника у коня). Маловажні такого значення не мають (гіпертермія при гастриті).

- постійні та непостійні. Постійні або стійкі реєструються обов'язково при даному захворюванні (відсутність акту дефекації за механічної непрохідності

кишківника; рідкісні та слабкі скорочення рубця у жуйних за гіпотонії передшлунків). Непостійні (нестійкі) - можуть протягом хвороби зникати (жовтяниця за гепатиту).

За локалізацією симптоми поділяються на:

- загальні (підвищення температури тіла, тахікардія, поліпное та ін.);

- місцеві (наявність тупого звуку під час перкусії легень, відкат задньої межі легень та ін.).

За прогнозом або передбаченням розрізняють симптоми:

- сприятливі (поява відрижки при тимпанії рубця, поява апетиту при гастроентериті). Свідчать про одужання тварини.

- несприятливі (поява ритму голопа за гострого міокардиту).

- загрозові (відсутність перистальтичних шумів у книжці). Передвіщають летальний результат.

- безнадійні (шум плескоти в серцевій сорочці за перикардиту), за яких тварини не одужують.

Розпізнавання симптомів - процес творчий і вимагає від лікаря великих знань з усіх клінічних дисциплін, а також досвіду і терпіння. На підставі тільки одного виявленого симптому розпізнати захворювання практично

неможливо. Найбільша кількість діагностичних помилок відбувається в результаті недостатнього дослідження. Тому найважливішим правилом діагностики є якомога повніше і систематичне вивчення хворої тварини.

Розпізнавання симптомів проводиться в кількох напрямках:

1) виявлені симптоми порівнюються з відповідними показниками, характерними для здорової тварини даного виду і віку;

2) симптоми зіставляються з такими, що описані в літературі за конкретного захворювання або синдрому;

3) симптоми класифікуються за цією схемою:

4) при повторному дослідженні оцінюють зміни виявлених раніше симптомів.

Оцінку діагностичної значущості симптомів проводять за ефектом лікувальних заходів, результатом хвороби, результатами патологоанатомічного розтину або анатомування тварини.

1.3.4 Діагноз хвороби та його класифікація

Діагноз - короткий лікарський висновок про сутність захворювання і стан тварини, виражений у нозологічних термінах. Діагноз не можна розглядати як щось закінчене і незмінне, оскільки він динамічний і може змінюватися залежно від перебігу хвороби, а також у результаті терапевтичних заходів.

Сформулювати або поставити діагноз це означає:

- визначити хворобливий процес;
- виявити функціональні та морфологічні зміни в тканинах, органах і системах;
- встановити причину захворювання.

Розрізняють діагноз хвороби та індивідуальний діагноз. Діагноз хвороби (*diagnosis morbi*) - визначають на підставі вивчення анамнезу і симптомів хвороби, які властиві всім, хто страждає на це захворювання. Прикладом такого діагнозу хвороби може слугувати пневмонія. Такий діагноз не розкриває всіх індивідуальних особливостей пацієнта, а хворі розглядаються як випадки, подібні один до одного.

Індивідуальний діагноз (*diagnosis aegroti*) відображає індивідуальні особливості перебігу хвороби в конкретній тварини в даний час і в даних умовах. Він ставиться тільки після ретельного дослідження пацієнта на підставі

клінічних, інструментальних, лабораторних та інших досліджень.

Діагноз, згідно з даним раніше визначенням, має бути виражений у нозологічних термінах. Під нозологічними термінами мають на увазі сукупність

слів або словосполучень, які є точним позначенням хворобливих процесів, що

розглядаються ветеринарною медициною. Нозологічна термінологія склалася в

процесі історичного розвитку ветеринарії та суміжних наук (медицини, мікробіології, вірусології, зоології тощо). При цьому нозологічна термінологія

постійно змінюється внаслідок відмирання, зникнення деяких старих і

виникнення нових хвороб тварин.

Прогноз (від грец. prognosis - передбачення) - передбачення розвитку і результату хвороби. Прогноз хвороби визначається на підставі:

- об'єктивних даних дослідження;

- знання закономірностей перебігу патологічних процесів;

- знання сутності хвороб;

- індивідуальних особливостей організму;

- наявності радикальних методів лікування;

- можливостей створення для тварини відповідних умов годування,

утримання тощо.

Прогноз може бути сприятливим, коли очікується повне одужання тварини зі збереженням продуктивності та працездатності. Несприятливий прогноз ставлять за невиліковних хвороб. Сумнівний або обережний прогноз

припускають, якщо складно точно визначити результат хвороби.

Саме на прогнозі ґрунтуються подальші ветеринарні та господарські заходи щодо хворої тварини або групи тварин. І насамкінець, за помилкового

прогнозу збитки несе власник тварини, який, цілком законно, може пред'явити матеріальні претензії до ветеринарного фахівця, авторитет якого також істотно

постраждає.

2. МОДЕЛЮВАННЯ СИСТЕМИ

2.1 Діаграма прецедентів

Діаграма прецедентів - це тип діаграми UML, який часто використовується для аналізу різних систем. Вони дають змогу візуалізувати різні типи ролей у системі та те, як ці ролі взаємодіють із системою.

Діаграми прецедентів використовуються для збору вимог до використання системи. Залежно від ваших вимог ви можете використовувати ці дані різними способами. Нижче наведено кілька способів їх використання.

Ідентифікація функцій і як з ними взаємодіють ролі - основне призначення діаграм сценаріїв використання.

Для представлення системи на високому рівні - Особливо корисно під час представлення керівникам або зацікавленим сторонам. Ви можете виокремити ролі, які взаємодіють із системою, і функціональні можливості, що надаються системою, не заглиблюючись у внутрішню роботу системи.

Ідентифікація внутрішніх і зовнішніх чинників - у великих складних проєктах система може бути ідентифікована як зовнішня роль в іншому випадку використання.

Об'єкти діаграми прецедентів

Використовувати діаграми прецедентів складаються з 4 об'єктів:

- Актор
- Випадок використання
- Система
- Пакет

Актор. У використовує діаграму прецедентів - це будь-яка сутність, яка виконує роль в одній даній системі. Це може бути людина, організація або зовнішня система.

Випадок використання. Являє собою функцію або дію всередині системи.

Система. Використовується для визначення сфери застосування і намальована у вигляді прямокутника. Це необов'язковий елемент, але корисний

під час візуалізації великих систем.

Пакет. Є ще одним додатковим елементом, який надзвичайно корисний у складних діаграмах. Подібно до діаграм класів, пакети використовуються для групування випадків використання.

На рис. 2 представлена діаграма прецедентів, що дозволяє ознайомитись з головними акторами в предметній області системи аналізу, а саме:

Клієнт :

- Перегляд сайту;

- Визначення захворювання;

- Реєстрація.

Лікар – співробітник лікарні, його задачі:

- Визначення захворювань;

- Надання рекомендацій по лікуванню;

Адміністратор – співробітник лікарні, має можливість:

- Вносити симптоми захворювань;

- Вносити захворювання та їх методи лікування;

- Надання прав лікаря користувачам.

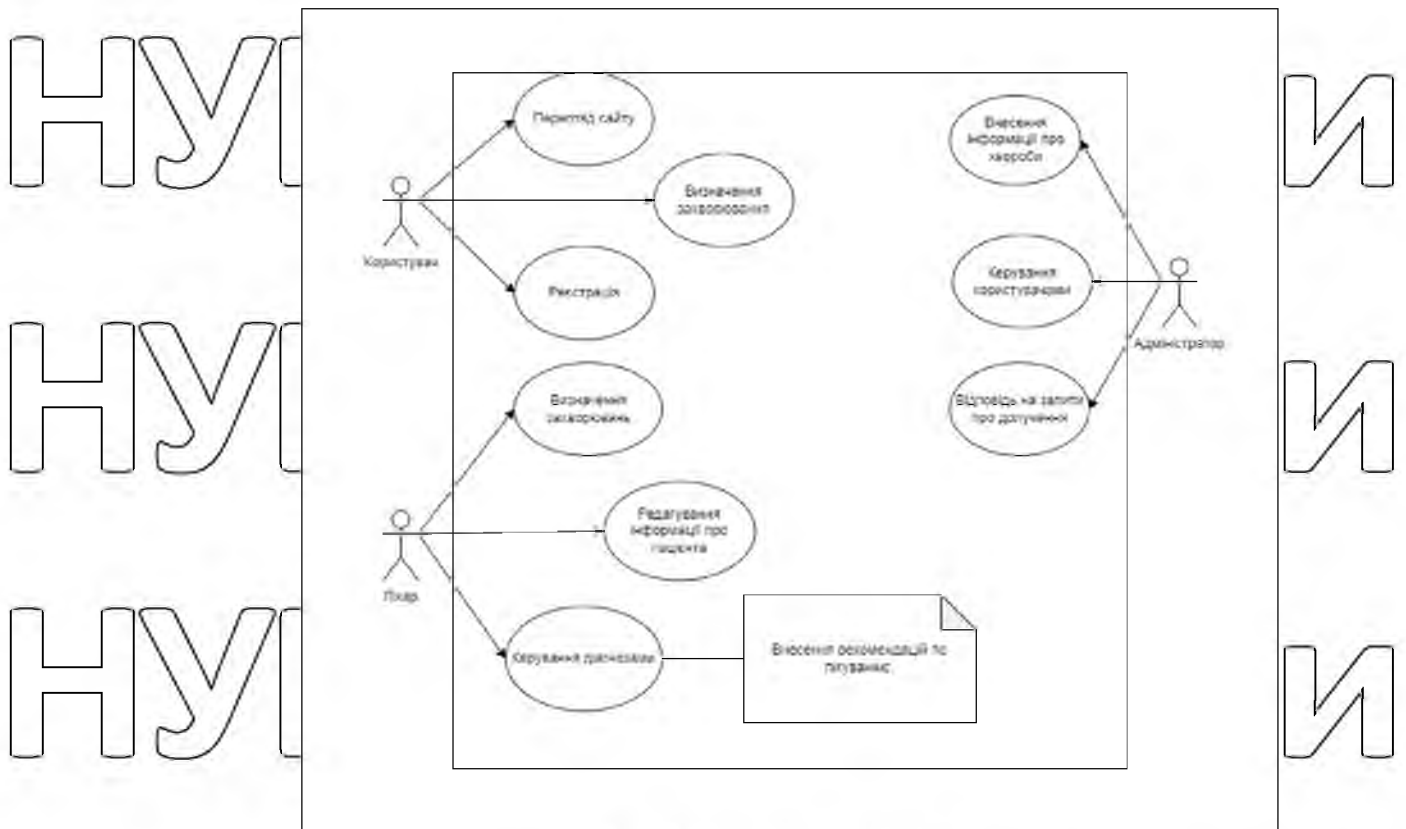


Рис. 2 Діаграма прецедентів

Після того, як ми визначили акторів і верхній рівень використання, у нас є базове уявлення про систему

2.2 SADT діаграма

SADT (Structured Analysis and Design Technique) - методологія аналізу та проектування систем.

Основні принципи методології SADT:

- Чіткий опис мети моделювання
- Фіксація єдиної точки зору на модельовану систему
- Визначення меж системи,
- Декомпозиція, що забезпечує послідовну деталізацію опису;
- Мета моделі - отримання відповідей на деяку сукупність запитань;
- Точка зору - це позиція спостерігача, яку необхідно вибрати, щоб побачити систему в дно;
- У моделі може бути тільки одна точка зору!

НУБІП України

- SADT модель повинна мати єдиний суб'єкт;
- У SADT-моделях використовуються як природна, так і графічна мови;
- Діаграма - сукупність графічного та відповідного текстового опису

Процес, діаграма якого побудована в нотації, що базується на принципах SADT (IDEFO, Процес, Процедура).

НУБІП України

Функціональна модель IDEFO являє собою набір блоків, кожен з яких являє собою "чорну шухляду" зі входами та виходами, управлінням і механізмами, які деталізуються (декомпонуються) до необхідного рівня. Найважливіша функція

НУБІП України

розташована у верхньому лівому кутку. А з'єднуються функції між собою за допомогою стрілок і описів функціональних блоків. При цьому кожен вид стрілки або активності має власне значення. Ця модель дає змогу описати всі основні види процесів, як адміністративні, так і організаційні. Стрілки можуть

бути:

НУБІП України

- Вхідні - вступні, які ставлять певне завдання.
- Вихідні - виводять результат діяльності
- Керуючі (зверху вниз) - механізми управління (положення, інструкції тощо).

НУБІП України

- Механізми (від низу до верху) - що використовується для того, щоб виконати необхідну роботу.

Розроблену IDEFO діаграму ми можемо побачити на рисунку 3

НУБІП України

НУБІП України

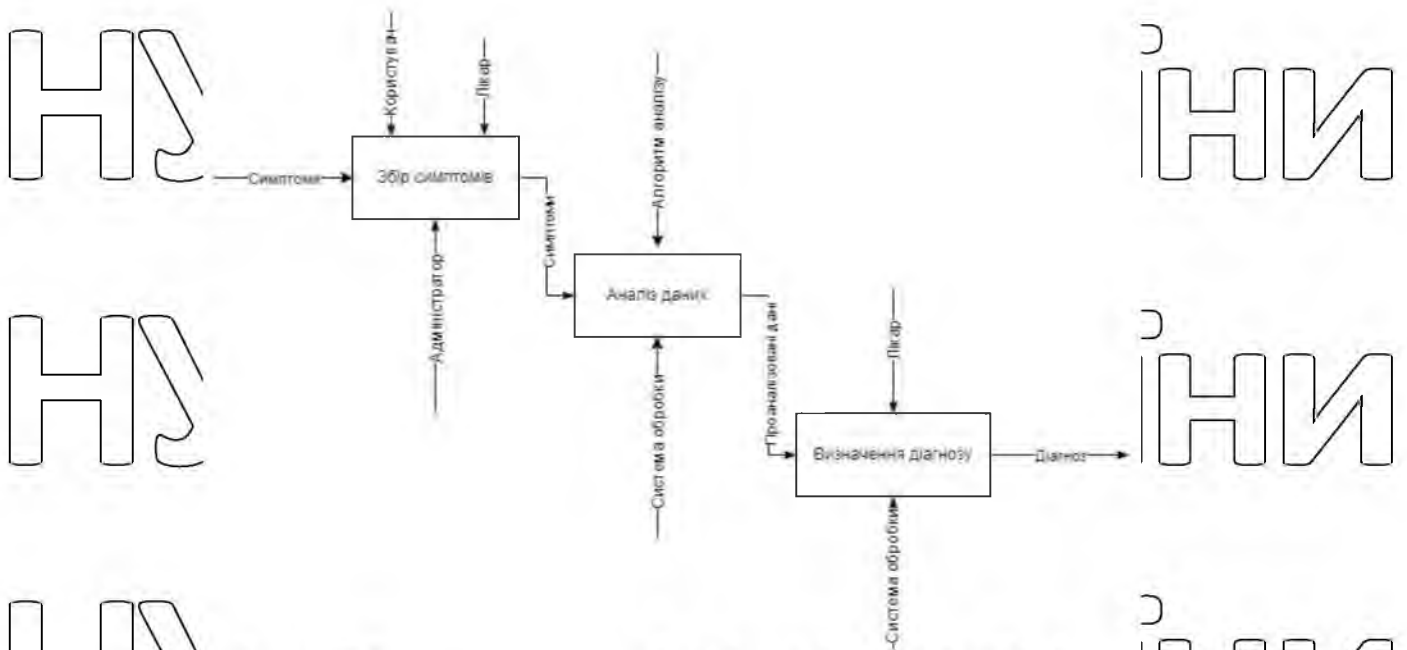


Рис. 3 Діаграма IDEFO

- Збір симптомів – перший блок відображає збір даних, користувач

та лікар вводять симптоми що є вхідним потоком.

- Аналіз даних – другий блок аналізу дані вибрані користувачем, вони збережені в системі та після цього представлені для аналізу.

- Визначення захворювання – останній блок діяльності, на основі проаналізованих даних визначає діагноз по симптомах які були проаналізовані системою.

2.2.1 Діаграма послідовності

За допомогою діаграми прецедентів (варіантів використання) виявляються основні користувачі системи та задачі, які дана система має вирішувати. За допомогою діаграми діяльності ми описуємо послідовність дій для кожного прецедента, необхідну для досягнення поставленої мети.

Далі проектується логічна структура системи за допомогою діаграми класів. На цьому етапі виокремлюють класи, що формують структуру БД Системи, а також класи, які реалізують певний набір операцій, що сприяє досягненню цілей у межах обраного прецеденту. Для опису складної поведінки деяких об'єктів (екземплярів класу) складається діаграма станів.

Таким чином, аналітики фіксують такі поведінкові аспекти, як алгоритм дій у межах одного або декількох прецедентів, необхідний для досягнення певного результату, а також зміна стану об'єктів під час виконання наведених дій.

Найчастіше на етапі специфікації вимог необхідно показати не тільки алгоритм дій або зміну стану об'єкта, а й обмін повідомленнями між окремими об'єктами Системи. Це завдання вирішує діаграма взаємодії.

Діаграма взаємодії призначена для моделювання відносин між об'єктами (ролями, класами, компонентами) Системи в рамках одного прецеденту.

Цей вид діаграм відображає такі аспекти проєктованої Системи:

- обмін повідомленнями між об'єктами (зокрема в межах обміну повідомленнями зі сторонніми Системами)
- обмеження, що накладаються на взаємодію об'єктів
- події, що ініціюють взаємодії об'єктів.

На відміну від діаграми діяльності, яка показує тільки послідовність (алгоритм) роботи Системи, діаграми взаємодії акцентують увагу розробників на повідомленнях, що ініціюють виклик певних операцій об'єкта (класу) або є результатом виконання операції.

Діаграма послідовності є одним із різновидів діаграм взаємодії та призначена для моделювання взаємодії об'єктів Системи в часі, а також обміну повідомленнями між ними.

Одним з основних принципів ООП є спосіб інформаційного обміну між елементами Системи, що виражається у надсиланні та отриманні повідомлень один від одного. Таким чином, основні поняття діаграми послідовності пов'язані з поняттям Об'єкт і Повідомлення.

Діаграму послідовності прецедента «Визначення захворювань» для експертної системи зображено в додатку А на сторінці 1.

2.3 Діаграма активності

Діаграми активностей (Activity Diagrams) є представленням алгоритмів деяких дій (активностей), що виконуються в системі. Нотація UML пропонує п'ять уявлень системи:

- Вигляд системи з точки зору прецедентів.
- Вигляд з точки зору проектування.
- Вигляд з точки зору процесів.
- Вигляд з точки зору розгортання.
- Вигляд з точки зору реалізації.

Можна побудувати кілька діаграм діяльності для однієї й тієї самої системи, до того ж кожна з них фокусуватиметься на різних аспектах системи, показуватиме різні дії, що виконуються всередині неї. Читач, звісно ж, зрозумів, що, коли ми говоримо про динаміку, ми маємо на увазі поведінку системи загалом або її частин. Говорячи більш формально, діаграми активності, загалом-то, не мають монополії на опис поведінкових особливостей динамічних частин системи. Для цієї ж мети можуть використовуватися ще діаграми прецедентів, послідовності, кооперації та станів.

Діаграму активності зображено на рисунку 4.

На ній ми можемо побачити процес визначення захворювання.

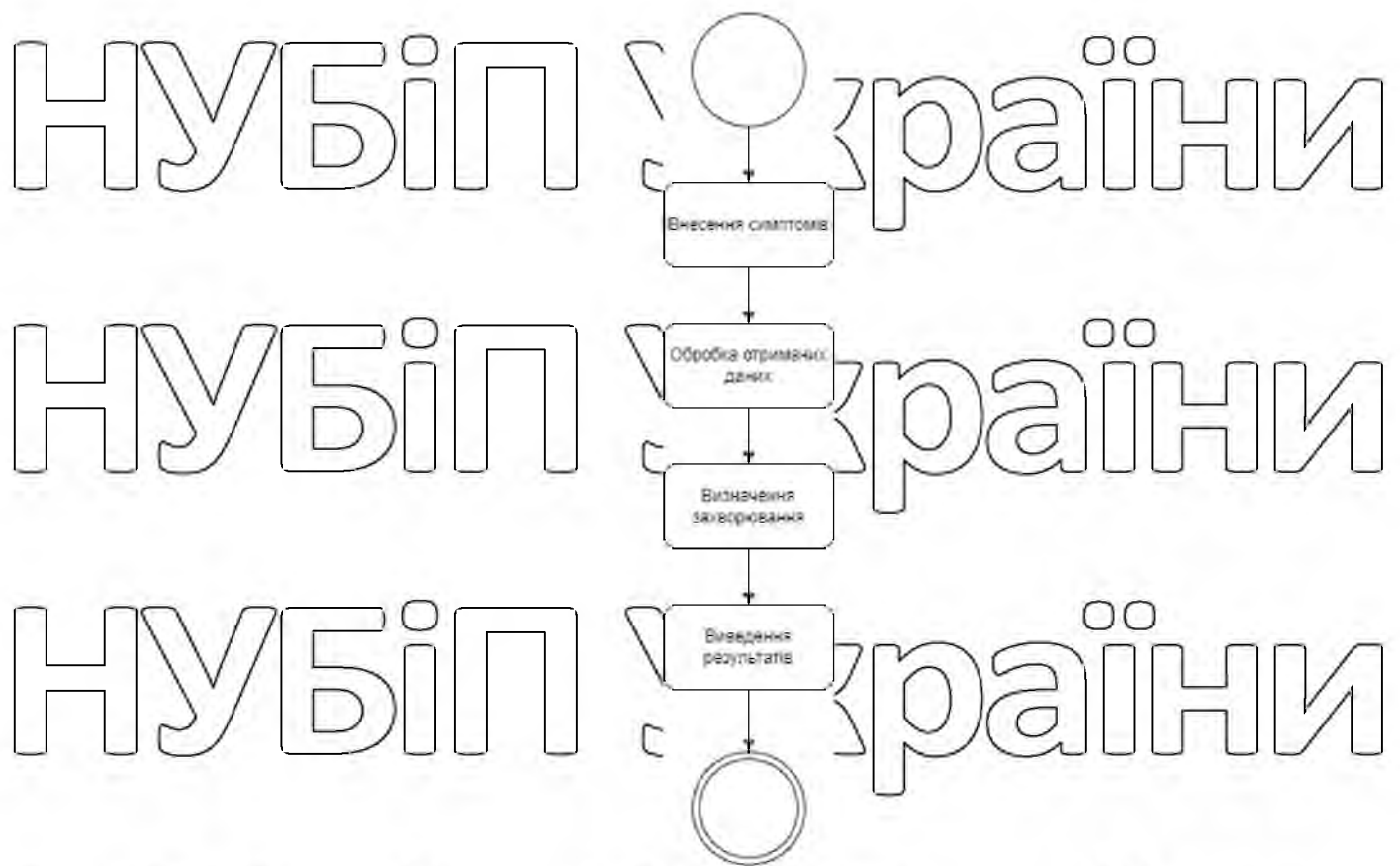


Рис. 4. Діаграма IDEF0

2.4 Діаграма кооперації

Діаграма кооперації - діаграма, на якій зображуються взаємодії між частинами композитної структури або ролями кооперації. На відміну від діаграми послідовності, на діаграмі комунікації явно вказуються відносини між об'єктами, а час як окремий вимір не використовується.

Головна особливість діаграми кооперації полягає в можливості графічно уявити не тільки послідовність взаємодії, а й усі структурні відносини між об'єктами, що беруть участь у цій взаємодії.

Насамперед на діаграмі кооперації у вигляді прямокутників зображуються об'єкти, які беруть участь у взаємодії, що містять ім'я об'єкта, його клас і, можливо, значення атрибутів. Далі, як і на діаграмі класів, вказуються асоціації між об'єктами у вигляді різних сполучних ліній. При цьому можна явно вказати імена асоціації та ролей, які відіграють об'єкти в даній асоціації. Додатково можуть бути зображені динамічні зв'язки - потоки повідомлень. Вони подані також у вигляді сполучних ліній між об'єктами, над якими розташовується

стрілка із зазначенням напрямку, імені повідомлення та порядкового номера в загальній послідовності ініціалізації повідомлень.

На рисунку 5. Представлена діаграма кооперації для системи визначення захворювань тварин. На ній ми можемо бачити що основні процеси відбуваються при взаємодії з системою. Об'єктами діграма є:

- Користувач
- Лікар
- Адміністратор
- Система захворювань тварин



Рис. 5 Діаграма кооперації

2.5 Діаграма діяльності

Діаграма діяльності - технологія, що дає змогу описувати логіку процедур, бізнес-процеси та потоки робіт

Основною відмінністю діаграм діяльності від блок-схем є активна підтримка паралельних процесів, що пояснює застосування діаграми діяльності для моделювання потоків робіт.

Кожен стан на діаграмі діяльності відповідає виконанню деякої

елементарної операції. Саме діаграми діяльності дають змогу реалізувати в мові UML особливості процедурного та синхронного керування. Метамоделю UML надає для цього необхідні терміни та семантику. Основним напрямом використання діаграм діяльності є візуалізація особливостей реалізації операцій класів, коли необхідно представити алгоритми їхнього виконання. При цьому кожен стан може бути виконанням операції деякого класу або її частини, даючи змогу використовувати діаграми діяльності для опису реакцій на внутрішні події системи

Діаграма діяльності до контрольного класу "Визначення захворювання"

відображена в додатку А на сторінці 2

Передумови.

Клієнт повинен бути зареєстрованим.

Головний потік.

Прецедент починає виконуватися, коли клієнт потрапляє на сторінку визначення захворювання. Система перевіряє введені дані (E-1) і виводить можливі варіанти отримання даних. Якщо вибрана операція S-1 отримання автоматичної відповіді системи або S-2 очікування коментаря лікаря, виконується потік виведення кінцевого діагнозу.

Під-потоки.

S-1: Отримання автоматичної відповіді. Система виводить автоматичний діагноз без коментаря лікаря

S-2: Очікування коментаря лікаря. Система очікує коментар лікаря і вже потім виводить кінцевий діагноз.

Альтернативні потоки

E-1: Введені неправильні дані. Користувач повинен зареєструватися або авторизуватися для доступу до визначення захворювання .

3. РОЗРОБКА СИСТЕМИ

НУБІП України

3.1 Архітектура системи

Архітектура системи - принципова організація системи, втілена в її елементах, їхніх взаємовідносинах між собою та з середовищем, а також принципи, що спрямовують її проєктування та еволюцію.

НУБІП України

Архітектура програмного забезпечення (англ. software architecture) -

сукупність найважливіших рішень про організацію програмної системи.

НУБІП України

Архітектура включає:

- вибір структурних елементів та їхніх інтерфейсів, за

допомогою яких складено систему, а також їхньої поведінки в

НУБІП України

рамках співпраці структурних елементів;

- з'єднання обраних елементів структури і поведінки, у все

більші системи,

- архітектурний стиль, який спрямовує всю організацію - всі

елементи, їхні інтерфейси, їхню співпрацю та їхнє з'єднання.

НУБІП України

Інформаційна архітектура (Enterprise Information Architecture, скор.

EIA), набір методик та інструментів, що описує інформаційну модель

підприємства. Включає:

НУБІП України

- бази даних і сховища даних;

- інформаційні потоки (як усередині організації, так і зв'язки із

зовнішнім світом).

НУБІП України

Архітектура прикладних рішень (Enterprise Solution Architecture

ESA) - представляє архітектуру застосунків, що містить сукупність

програмних продуктів та інтерфейсів між ними. Ділиться на два напрямки:

- область розроблення прикладних систем;
- портфель прикладних систем

Технічна архітектура (Enterprise Technical Architecture скор. ЕТА) – сукупність програмно-апаратних засобів, методів і стандартів, що

забезпечують ефективне функціонування додатків. Описує повне уявлення інфраструктури підприємства, включно з:

- інформацію про інфраструктуру підприємства;
- системне програмне забезпечення (СУБД, системи інтеграції);
- стандарти на програмно-апаратні засоби;
- засоби забезпечення безпеки (програмно-апаратні);
- системи управління інфраструктурою.

В проектуванні системи була використана технічна архітектура. Архітектура система зображено на (рис. 6) вона складається з таких частин:

- сервер, використовується для розміщення веб-системи з веб-платформою, яка надає можливість користувачу системи визначити захворювання свійської тварини;
- пристрій користувача з допомогою якого він зв'язується з сервером та може використовувати веб систему;
- база даних, зберігає всю інформацію про користувачів та результати досліджень

- сервер, місце де зберігається база даних і веб система, виконує аналіз даних та генерує результати досліджень;
- Таблиця БД «zoo», місце зберігання результатів аналізу даних.

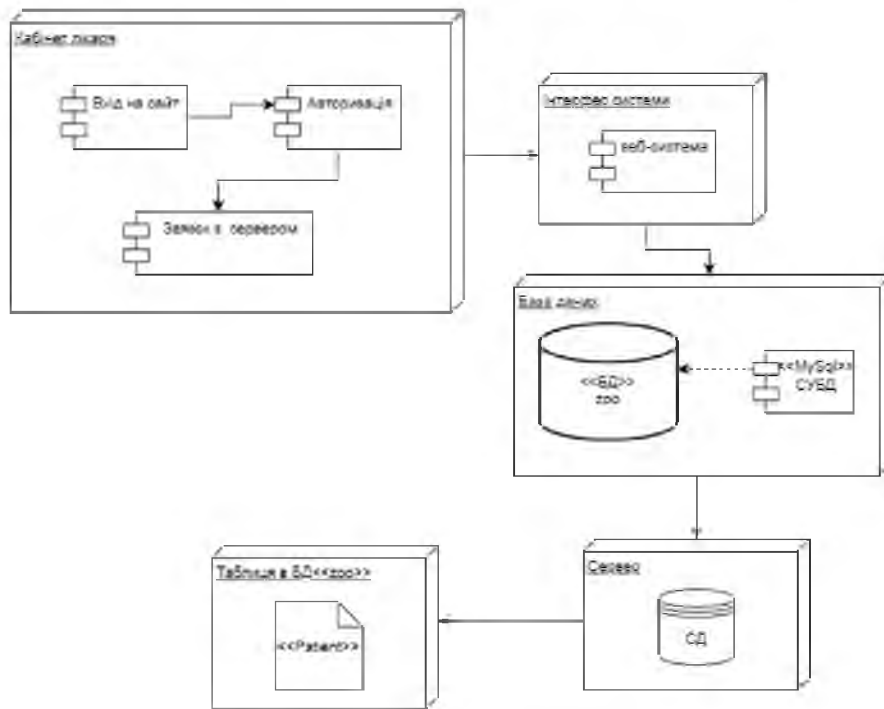


Рис. 6 Топологія системи

3.2 Інформаційне забезпечення

3.2.1 Операційна база даних.

Система управління базами даних (СУБД, Data Base Management System) – це сукупність програмних засобів, призначених для створення та використання баз даних.

Основні завдання СУБД:

- надавати користувачам доступ до даних без знання їхньої фізичної організації;
- забезпечувати безпеку зберігання даних і доступу до них;
- забезпечувати безконфліктність одночасного доступу різних користувачів до одних даних;
- реалізовувати транзакції.

СУБД розрізняються:

- за підтримкою моделей даних (реляційні, ієрархічні, мережеві, об'єктні,...)
- за способом доступу до БД (файл-серверні, клієнт-серверні, вбудовані)

- за масштабами підтримуваних баз
- за платформами реалізації

Реляційна модель - це модель "сутність-зв'язок". Реляційну базу даних можна розглядати як колекцію таблиць, пов'язаних між собою.

- Переваги реляційної моделі:
- універсальність;
- відносна простота реалізації;
- підтримка стандартним інтерфейсом SQL.

Незалежно від фізичної організації, таблицю можна розглядати як набір записів і полів. Записи таблиці містять відомості про однотипні об'єкти, що мають стандартний набір параметрів.

Дані в таблицях повинні задовольняти таким принципам:

- Кожне значення на перетині рядка і стовпця атомарне.
- Значення даних в одному й тому самому стовпці належать до одного й того самого типу даних.
- Кожен запис у таблиці унікальний (тобто в таблиці не може бути двох і

- більше повністю співпадаючих рядків).
- Кожен стовпець має унікальне ім'я.
- Послідовність записів у таблиці та послідовність полів у записі несуттєва.

При розробці системи було виявлено 5 основних сутностей:

- **Діагноз** - сутність зберігає дані про визначені захворювання та його симптоми.

НУБІП України

- **Симптоми** – сутність зберігає назву.
- **Захворювання** – сутність зберігає текстову інформацію, його назву, назву та рекомендацію по лікуванню
- **Користувач** – сутність зберігає інформацію про користувача його, логін, пароль та контакти.
- **Запити** – сутність зберігає інформацію про надіслані запити на визначення захворювання, симптоми захворювання, відповідь лікаря та його рекомендації по лікуванню.

Логічна модель даних зображена на рисунку 7.

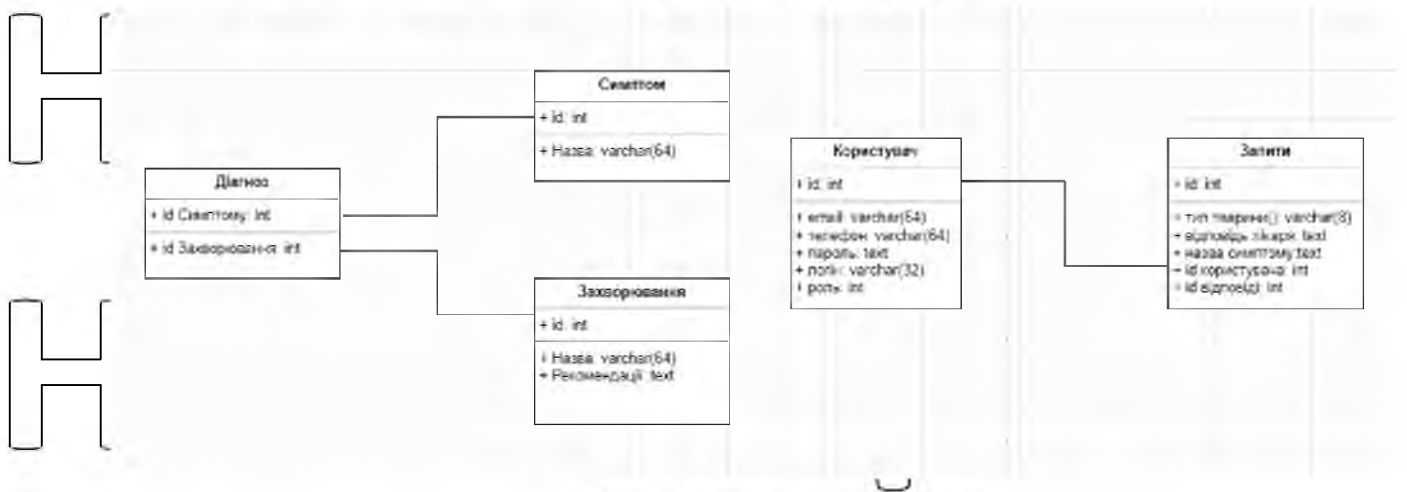


Рис. 7 Логічна модель даних

Для створення таблиць використовується функція «CREATE TABLE» після якої, вказується назва нової таблиці. Далі вказуються значення таблиці та їх типи. NOT NULL вказує на те, що значення не може бути пустим. PRIMARY KEY використовується для однозначної ідентифікації рядків в таблиці. Значення первинного ключа має бути завжди унікальним і не повторюватись. Приклад створення таблиці application та diseases зображено на рисунках 8 та 9.

НУБІП України

```

CREATE TABLE `applications` (
  `id` int NOT NULL,
  `pet_type` varchar(8) DEFAULT NULL,
  `response_doctor` text,
  `symptoms_ids` text,
  `user_id` int DEFAULT NULL,
  `is_response_send` int DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

```

Рис. 8 Створення таблиці «application»

```

CREATE TABLE `diseases` (
  `id` int NOT NULL,
  `title` varchar(64) DEFAULT NULL,
  `recommendations` text
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

```

Рис. 9 Створення таблиці «diseases»

Для поточної системи було створено 5 таблиць. Запити на створення всіх таблиць можна побачити у додатку А.

3.2.3 Структура сховища даних.

Сховище даних - це різновид системи управління даними, яка забезпечує підтримку бізнес-аналітики. Сховища даних призначені тільки для виконання запитів і аналізу та зазвичай містять великі обсяги історичних даних. Дані зазвичай надходять у сховище з найрізноманітніших джерел, як-от журнали додатків і додатки транзакцій.

Сховище даних служать для централізації та консолідації великих обсягів даних з різних джерел. Аналітичні інструменти дають можливість компаніям витягувати з власних даних цінні для бізнесу відомості та підвищувати ефективність прийнятих рішень. Згодом у сховищі накопичуються записи за минулі періоди, які становлять велику цінність для фахівців з вивчення даних і бізнес-аналітиків. Ці можливості роблять сховища даних єдиним джерелом перевіреної інформації компанії.

Головним етапом створення системи є створення сховища даних (СД), наповнення якого буде відбуватись за рахунок оперативної БД. На рисунку 16

представлено створене СД.

Сховище даних містить 5 таблиць. Структуру сховища даних ми можемо побачити на рисунку 10.

НУБІП України

Н
Н

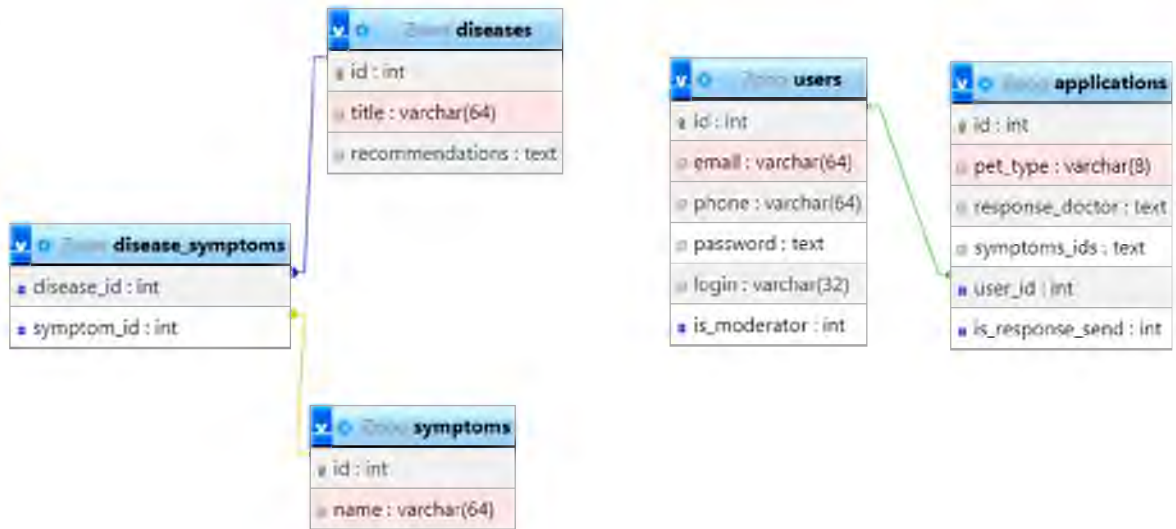


Рисунок 10. Структура сховища даних

НУБІП України

Інформація в таблицях вимірів є відносно постійною, тож дані цих таблиць заповнювалися за допомогою SQL запитів, приклад якого наведено на рисунку

11.

```
INSERT INTO `diseases` (`id`, `title`, `recommendations`) VALUES  
(1, 'Грип птиці', 'Для специфічної профілактики використовують інактивовані вакцини.'  
(2, 'Бруцельоз', 'Оптимальне лікування вимагає призначення 2 антибіотиків - доксицикл  
(3, 'Некробактеріоз', 'Лікування некробактеріозу ВРХ, свиней, овець Хвору тварину із  
(4, 'Мікоплазмоз', 'Лікування уrogenітального мікоплазмозу зводиться до прийому антиб  
(5, 'Актиномікоз', 'Терапію проводять антибактеріальними препаратами. Зазвичай вибір  
(6, 'Лептоспіроз', 'Лептоспіроз лечат антибиотиками. Но антибиотикотерапия (пеницилли
```

НУБІП України

Рисунок 11. Приклад SQL запиту на занесення даних

Виконавши всі необхідні запити, можна перевірити занесені дані Приклад занесених даних відображено на рисунку 12.

НУБІП України

	id	title	recommendations
1	Грип птиці	Для специфічної профілактики використовують інакти...	
2	Бруцельоз	Оптимальне лікування вимагає призначення 2 антибіо...	
3	Некробактеріоз	Лікування некробактеріозу ВРХ, свиней, овець Хвору...	
4	Мікоплазмоз	Лікування уrogenітального мікоплазмозу зводиться д...	
5	Актиномікоз	Терапію проводять антибактеріальними препаратами. ...	
6	Лептоспіроз	Лептоспіроз лечат антибиотиками. Но антибиотикотер...	

Рисунок 12. Приклад занесених даних

НУБІП України

3.3 Засоби розробки Web-сайту

3.3.1 Мова розмітки гіпертекстових сторінок HTML

Мова розмітки гіпертекстових сторінок (HTML - Hypertext Markup Language) являє собою мову, розроблену спеціально для створення Web-документів. Вона визначає синтаксис і розміщення спеціальних інструкцій

(тегів), які не виводяться на екран, але вказують браузеру, як відображати вміст документа. Він також використовується для створення посилань на інші документи, локальні або мережеві, наприклад ті, що знаходяться в мережі Інтернет.

Документ HTML містить текст (вміст сторінки) і вбудовані теги - інструкції про структуру, зовнішній вигляд і функції вмісту. Документ HTML поділяється на дві основні частини: заголовок - head і тіло - body. Заголовок містить такі відомості про документ, як його назва та методична інформація, що описує вміст. У тілі міститься сам вміст документа (те, що виводиться у вікні браузера).

Приклад коду:

```
<div class="our-services">
<div class="header-our-services">
```

`<h2 class="title">`
НУБІП України
`</h2>`

Our Services & Procedures

`<p class="info-title-about">`COMPASSIONATE AND HIGH
QUALITY CARE`</p>`

`</div>`
НУБІП України
`<div class="card-for-us">`
`</div>`

`<h4>`Wellness Care`</h4>`

`<p>`Our wellness programs include: comprehensive physical exam;
internal and external parasite testing.
`</p>`
`</div>`

`<div>`

`<h4>`Anesthetic Monitoring`</h4>`
НУБІП України

`<p>`Your pet will be examined during check in to ensure that he or she
is in top shape for surgery that

day.`</p>`

3.3.2 Каскадна таблиця стилів CSS
НУБІП України

CSS - формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки.

CSS використовується для завдання кольорів, шрифтів, розташування окремих блоків та інших аспектів подання зовнішнього вигляду web-сторінок. Основною метою розроблення CSS був поділ опису логічної структури web-сторінки (який здійснюють за допомогою html або інших мов розмітки) від опису зовнішнього вигляду цієї web-сторінки (який тепер здійснюють за допомогою формальної мови CSS). Такий поділ може збільшити доступність документа, надати більшу гнучкість і можливість керування його поданням, а також зменшити складність і повторюваність у структурному вмісті.

До появи CSS оформлення web-сторінок здійснювалося виключно засобами html, безпосередньо всередині вмісту документа. Однак із появою CSS стало можливим принципове розділення змісту та подання документа. За рахунок цього нововведення стало можливим легке застосування єдиного стилю оформлення для маси схожих документів, а також швидка зміна цього оформлення.

Переваги:

- кілька дизайнів сторінки для різних пристроїв перегляду.

Наприклад, на екрані дизайн буде розрахований на більшу ширину,

під час друку меню не буде виводитися, а на КПК і стільниковому телефоні меню буде слідувати за вмістом;

- зменшення часу завантаження сторінок сайту за рахунок

перенесення правил подання даних в окремий css-файл. У цьому

разі браузер завантажує тільки структуру документа і дані, що

зберігаються на сторінці, а представлення цих даних завантажується браузером тільки один раз і може бути закешоване;

- простота подальшої зміни дизайну. Не потрібно правити кожен сторінку, а лише змінити CSS-файл;

- додаткові можливості оформлення. Наприклад, за допомогою css-верстки можна зробити блок тексту, який рента тексту буде обтікати (наприклад, для меню) або зробити так, щоб меню було завжди видно під час прокрутки сторінки.

Приклад коду:

```
a {  
text-decoration: none;  
}
```

```
body, html {  
padding: 0;  
margin: 0;
```

```
color: #606060;  
font-family: 'fons';
```

НУБІП Україна

```
.wrap-header {  
display: flex;  
justify-content: space-between;  
width: 95%;  
margin: auto;
```

НУБІП Україна

```
padding: 10px 0px;  
align-items: center;
```

НУБІП Україна

```
.contact {  
display: flex;
```

НУБІП Україна

3.3.3 Препроцесор гіпертексту PHP

Препроцесор гіпертексту (PHP - Hypertext Preprocessor) - це широко використовувана мова сценаріїв загального призначення з відкритим вихідним кодом.

НУБІП Україна

Синтаксис мови бере початок із C, Java і Perl. PHP досить простий для вивчення. Перевагою PHP є надання web-розробникам можливості швидкого створення динамічно генерованих web-сторінок.

НУБІП Україна

Значною відмінністю PHP від будь-якого коду, що виконується на стороні клієнта, наприклад, JavaScript, є те, що PHP-сценарії виконуються на стороні сервера. Можна сконфігурувати свій сервер таким чином, щоб html-файли оброблялися процесором PHP, так що клієнти навіть не зможуть дізнатися, чи отримують вони звичайний html-файл або результат виконання сценарію.

НУБІП Україна

PHP дає змогу створювати якісні Web-додатки за дуже короткі терміни, отримуючи продукти, які легко модифікувати і підтримувати в майбутньому.

Головним фактором мови PHP є практичність. PHP має надати програмісту засоби для швидкого та ефективного вирішення поставлених завдань. Практичний характер PHP обумовлений п'ятьма важливими характеристиками:

- традиційністю;
- простотою;
- ефективністю;
- безпекою;
- гнучкістю.

Далі ми можемо бачити код створення нових симптомів:

```
<div class="add-symp">
  <select name="add_symptom_disease" id="<?=$disease[id]?="
class="add_symptom_disease">
    <?php foreach ($disease['available_symptoms'] as
$available_symptom);?>
      <option value="<?=$available_symptom[id]?="><?=$
$available_symptom[name]?></option>
    <?php endforeach;?>
  </select>
  <button class="add_symptom_admin" id="<?=$
$disease[id]?=">Add symptom</button>
</div>
```

3.3.4 SQL

Мова структурованих запитів (SQL) - це мова програмування для зберігання та обробки інформації в реляційній базі даних. Реляційна база даних зберігає інформацію в табличній формі з рядками і стовпчиками, що представляють різні атрибути даних і різні зв'язки між значеннями даних.

Інструкції SQL можна використовувати для зберігання, оновлення, видалення, пошуку та вилучення інформації з бази даних. Можна також використовувати SQL для підтримки та оптимізації продуктивності бази даних.

Мова структурованих запитів (SQL) - популярна мова запитів, яка часто

використовується в усіх типах додатків. Аналітики даних і розробники вивчають і використовують SQL, тому що це рішення добре інтегрується з різними мовами програмування. Наприклад, вони можуть впроваджувати SQL-запити з мовою програмування Java для створення високопродуктивних додатків обробки даних з основними системами баз даних SQL, такими як Oracle або MS SQL Server.

Рішення SQL також досить просте в освоєнні, оскільки в його твердженнях використовуються загальноприйняті англійські ключові слова.

Рішення SQL було винайдено в 1970-х роках на основі реляційної моделі даних. Спочатку воно було відоме як структурована англійська мова запитів (SEQUEL). Пізніше цей термін було скорочено до SQL. Компанія Oracle, раніше - Relational Software, стала першим постачальником, який запропонував комерційну систему управління реляційними базами даних SQL.

Системи управління реляційними базами даних використовують мову структурованих запитів (SQL) для зберігання даних і управління ними. У системі зберігається кілька таблиць бази даних, пов'язаних одна з одною. MS SQL Server, MySQL або MS Access є прикладами систем керування реляційними базами даних. Нижче перелічено компоненти такої системи.

Таблиця SQL - це базовий елемент реляційної бази даних. Таблиця бази даних SQL складається з рядків і стовпців. Інженери баз даних створюють зв'язки між кількома таблицями бази даних, щоб оптимізувати простір для зберігання даних.

НУБІП України

НУБІП України

3.4 Тестування системи

На головній сторінці сайту користувач може ознайомитися з сайтом, переглянути інформацію про нього. Для користувача доступні такі сторінки.

- Онлайн підтримка

- Контакти
- Акаунт
- Симптоми

Для незаєєстровано користувача доступ до сторінки визначення захворювання закритий при переході на неї з'являється вікно з вказівкою про реєстрацію.

Онлайн підтримка це сторінка з посиланням на телеграм канал в якому користувач може спілкуватися з лікарем та отримувати більшш детальну інформацію по лікуванню. Перехід в телеграм канал зображено на рисунку

Рисунок 13



Рисунок 13. Улюблені страви користувача

На сторінці Акаунт користувач може переглянути результати визначення діагнозу та рекомендації по лікуванню хвороби.

На рисунку 14 ми можемо ознайомитися з головною сторінкою сайту на ній розміщено головне меню сайту та кнопка реєстрації або входу на сайт



About Clinic

Рисунок 14. Головна сторінка

На рисунку 15 зображено приклад відгуків і оцінок на певну страву.



Рисунок 15. Реєстрація вхід в особистий кабінет

Після реєстрації або входу в обліковий запис клієнт може зробити запит на визначення захворювання. Для цього йому потрібно з випадальною списку вибрати симптоми та для того щоб лікарю було більш зрозуміло з якою твариною він має справу, вибір домашня чи свійська. Це полегшить йому написання рекомендацій по лікуванню. Зображено на рисунку 16

НУБІП УКРАЇНИ



Рисунок 16. Визначення діагнозу

Після визначення захворювання при переході в свій кабінет користувач може побачити результати дослідження, рисунок 17.



Рисунок 17. Результати дослідження

В панелі адміністратора доступні функції:

- Додавання симптомів
- Внесення захворювань
- Надання прав користування

На рисунку 18 ми можемо бачити сторінку надання прав користування сайтом. Кожен користувач може зв'язатися з адміністратором та запросити доступ до прав лікаря. Після чого адміністратор за допомогою панелі керування



Рисунок 18. Доступ користувачів

Для поповнення бази хворіб адміністратор має форму заповнення даних в якій для оновлення вибирає наявні симптоми та в залежності від захворювання вносить зображено на рисунку 19.

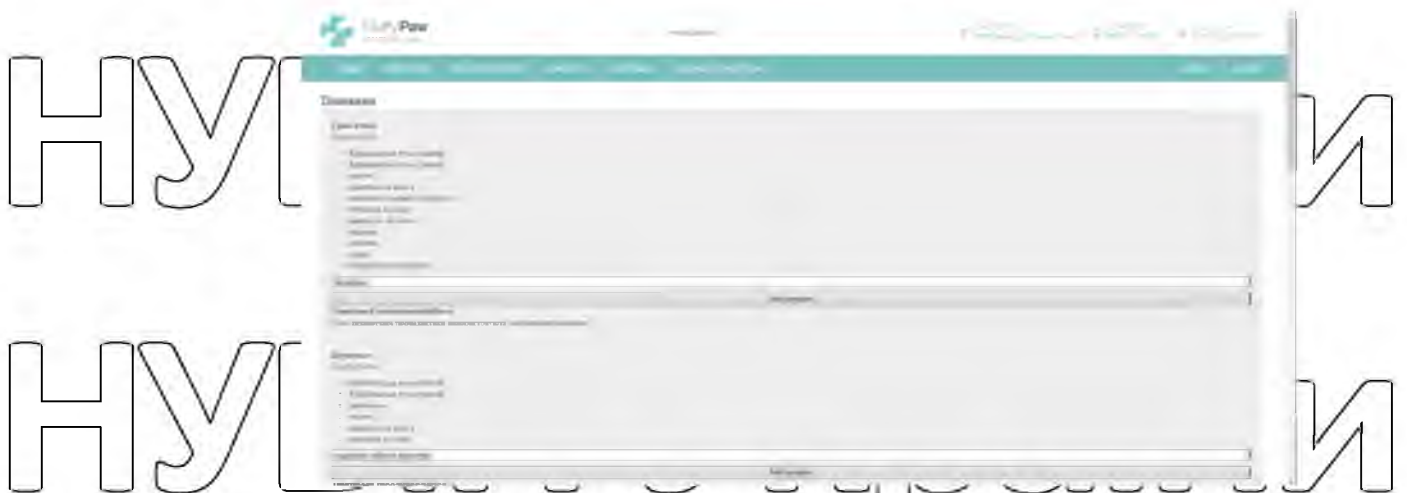


Рисунок 19. База хворіб

Сторінка симптоми зображена на рисунку 20. Для того щоб створювати нові хвороби потрібно мати симптоми. На цій сторінці відображено форму введення назв симптомів які потім використовуються для діагностування хворіб. При аналізі предметної області були вибрані загальні симптоми для більшості хворіб що спрощує діагностування більшості захворювань.



Рисунок 20. Сторінка симптоми

НУБІП України

На сторінці лікаря рисунок 21 відображуються сформовані системою діагнози. Лікар може ознайомитися з симптомами та побачити визначену хворобу. Після чого може додати свій коментар та рекомендації по лікуванню хвороби.

НУБІП України

НУБІП України

НУБІП України

НУБІП України



Рисунок 21. Сторінка лікаря

НУБІП України

НУБІП України

НУБІП України

НУБІП України

ВИСНОВКИ

У ході виконання магістерської роботи було проведено аналіз предметної області, визначено основні елементи та модулі, які в подальшому необхідні для створення експертної системи, змодельована та розроблена інформаційна система у вигляді веб-системи.

Система була розроблена користуючись мовою розмітки HTML/CSS, серверною мовою програмування PHP, бібліотекою JQuery та додатковими CSS фреймворками Bootstrap та UIKit.

Результатом даної роботи є готова експертна система на основі маркетингових досліджень ринку ресторанних послуг яка дозволить покращити конверсію продажів тим самим покращити прибутковість та впізнаваність.

Також система дозволяє автоматизувати прийом замовлень у ресторані при двох варіантів замовлення (замовлення у ресторані та замовлення доставки), автоматизувати роботу працівників ресторану та надає змогу вести можливість обліку продажу страв, отримання фідбеку від клієнтів та деякі маркетингові можливості.

СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Структурно-функціональний аналіз системи / [Електронний ресурс] - Режим доступу:

http://bookwu.net/book_metodologiya-sistemnogo-pidhodu_996/6_2.2.-sutnist-principi-osnovni-etapi-ta-metodi-sistemnogo-analizu

2. Тернопільський державний економічний університет [Електронний ресурс]: «Організація баз даних» - Режим доступу:

<http://dspace.wunu.edu.ua/jspui/bitstream/316497/9109/1/%D0%BE%D0%BF%D0%BE%D1%80%D0%BD%D0%B8%D0%B9%20%D0%BA%D0%BE%D0%BD%D1%81%D0%BF%D0%B5%D0%BA%D1%82%20%D0%BB%D0%B5%D0%BA%D1%86%D1%96%D0%B9.pdf>

3. Основи SQL. Навч. посібн. / Б.М. Дринь, О.В. Козич - Івано-Франківськ, Прикарпатський національний університет, ВДВ ЦІТ "Плай", 2012. - с.

4. ScikitLearn / [Електронний ресурс]: «Decision Trees» - Режим доступу:

<https://scikit-learn.org/stable/modules/tree.html>

5. ЕКСПЕРТНА СИСТЕМА ДЛЯ ДІАГНОСТИКИ ЗАБОЛЕВАНЬ КОРОВ / [Електронний ресурс]: «Naive Bayes» - Режим доступу:

<https://kazatu.edu.kz/assets/i/science/sf16-ikt-320.pdf>

6. Вескес Л.Дж. Access i SQL Server / Керівництво розробника / Дж. Л. Вескес - Москва: Уорі, 1997. Garcia, Enrique (2007) [Електронний ресурс]: «Drawbacks and solutions of applying association rule mining in learning management systems» - Режим доступу:

<https://sei2s.ugr.es/keel/pdf/specific/congreso/3-associationrules-Final.pdf>

7. Precisely / [Електронний ресурс]: «Техніки аналізу даних: 5 найкращих для розгляду» - Режим доступу:

<https://www.precisely.com/blog/datagovernance/top-5-data-mining-techniques>

8. Talend - A Leader in Data Integration & Data Integrity / [Електронний ресурс]: «16 Data Mining Techniques: The Complete List - Talend» - Режим доступу:

<https://www.talend.com/resources/data-mining-techniques/>

9. SearchBusinessAnalytics / [Електронний ресурс]: «What are Association

Rules in Data Mining (Association Rule Mining)?» Режим доступу:
<https://searchbusinessanalytics.techtarget.com/definition/association-rules-in-data-mining>

НУБІП УКРАЇНИ

10. К. Дж. Дейт - SQL и реляционная теория. Как грамотно писать код на SQL. 2008.

11. Программы для сельского хозяйства - Режим доступу:
https://www.kcrall-agro.ru/tree_diag_an.htm

НУБІП УКРАЇНИ

12. ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА / [Електронний ресурс]: «Експертні системи» - Режим доступу:

https://financial.lnu.edu.ua/wp-content/uploads/2019/09/ES_konspekt-veksy.pdf

НУБІП УКРАЇНИ

НУБІП УКРАЇНИ

НУБІП УКРАЇНИ

НУБІП УКРАЇНИ

НУБІП УКРАЇНИ

НУБІП України

НУБІП України

НУБІП України

ГРАФІЧНІ ДОДАТКИ

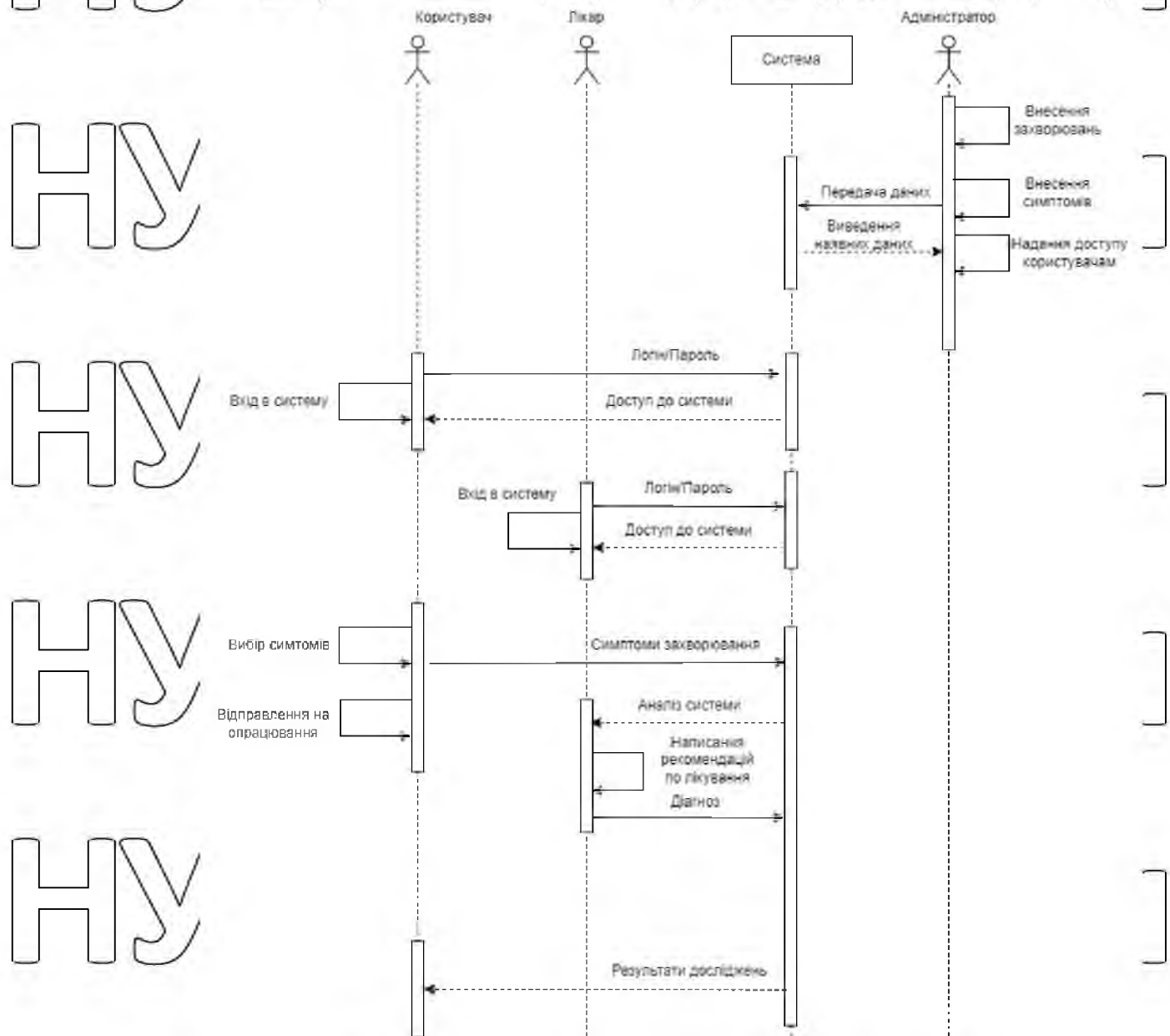
НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України



НУБІП України

НУБІП України

НУБІГ

НУБІГ

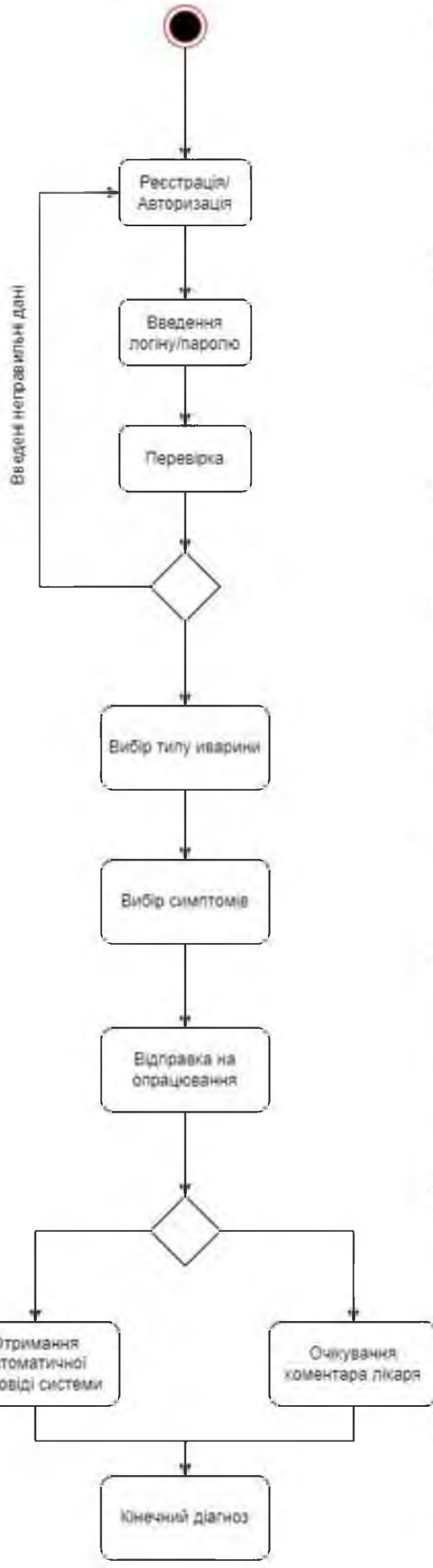
НУБІГ

НУБІГ

НУБІГ

НУБІГ

НУБІГ



ІНІИ

ІНІИ

ІНІИ

ІНІИ

ІНІИ

ІНІИ

ІНІИ

ДОДАТОК Б
НУБІП України

НУБІП України

НУБІП України

СТВОРЕННЯ БАЗИ ДАНИХ

НУБІП України

НУБІП України

НУБІП України

Сторінок 5
НУБІП України

НУБІП України

```

DROP TABLE IF EXISTS `applications`;
CREATE TABLE `applications` (
  `id` int NOT NULL AUTO_INCREMENT,
  `pet_type` varchar(8) DEFAULT NULL,
  `response_doctor` text,
  `symptoms_ids` text,
  `user_id` int DEFAULT NULL,
  `is_response_send` int DEFAULT '0',
  PRIMARY KEY (`id`),
  KEY `user_id` (`user_id`),
  CONSTRAINT `applications_ibfk_1` FOREIGN KEY (`user_id`)
REFERENCES `users` (`id`)
);

```

```

LOCK TABLES `applications` WRITE;
UNLOCK TABLES;

```

НУБІП України

```

DROP TABLE IF EXISTS `disease_symptoms`;
CREATE TABLE `disease_symptoms` (
  `disease_id` int DEFAULT NULL,
  `symptom_id` int DEFAULT NULL,
  KEY `disease_id` (`disease_id`),
  KEY `symptom_id` (`symptom_id`),
  CONSTRAINT `disease_symptoms_ibfk_1` FOREIGN KEY
(`disease_id`) REFERENCES `diseases` (`id`),
  CONSTRAINT `disease_symptoms_ibfk_2` FOREIGN KEY
(`symptom_id`) REFERENCES `symptoms` (`id`)
);

```

```

LOCK TABLES `disease_symptoms` WRITE;
UNLOCK TABLES;

```

НУБІП України

```

DROP TABLE IF EXISTS `diseases`;
CREATE TABLE `diseases` (
  `id` int NOT NULL AUTO_INCREMENT,
  `title` varchar(64) DEFAULT NULL,
  `recommendations` text,
  PRIMARY KEY (`id`)
);

```

```

LOCK TABLES `diseases` WRITE;
UNLOCK TABLES;

```

НУБІП України

```

DROP TABLE IF EXISTS `symptoms`;
CREATE TABLE `symptoms` (
  `id` int NOT NULL AUTO_INCREMENT,
  `name` varchar(64) NOT NULL,
  PRIMARY KEY (`id`)
);

```

```

LOCK TABLES `symptoms` WRITE;
UNLOCK TABLES;

```

НУБІП України

```

DROP TABLE IF EXISTS `users`;
CREATE TABLE `users` (
  `id` int NOT NULL AUTO_INCREMENT,
  `email` varchar(64) NOT NULL,
  `phone` varchar(64) NOT NULL,
  `password` text NOT NULL,
  `login` varchar(32) DEFAULT NULL,
  `is_moderator` int DEFAULT '0',

```

НУБІП України

PRIMARY KEY (`id`)
);
LOCK TABLES `users` WRITE;
UNLOCK TABLES;
НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України